

# Capacitor : from PWA to native app in 5 minutes

Horacio Gonzalez @LostInBrittany



@LostInBrittany #Capacitor #DevFestLille

# Who am I?

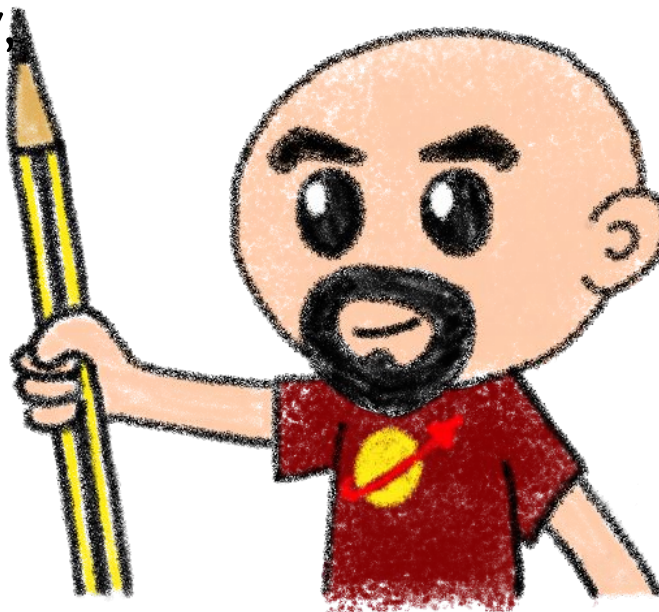
Introducing myself and  
introducing OVH



# Horacio Gonzalez

@LostInBrittany

Spaniard lost in Brittany,  
developer, dreamer and  
all-around geek



# OVH : Key Figures



**1.3M** Customers worldwide in **138** Countries

**1.5 Billions euros** investment over five years

**30** Datacenters (growing)

**350k** Dedicated Servers

**200k** Private cloud VMs running

**650k** Public cloud Instances created in a month

**15TB** bandwidth capacity

**35** Points of presence

**4TB** Anti DDoS capacity

Hosting capacity : **1.3M** Physical Servers

+ **2 500** Employees in **19** countries  
**20** Years of Innovation



# Ranking & Recognition



1<sup>st</sup> **European** Cloud Provider\*

1<sup>st</sup> **Hosting** provider in Europe

1<sup>st</sup> **Provider** Microsoft Exchange

**Certified** vCloud Datacenter

**Certified** Kubernetes platform (CNCF)

Vmware **Global Service Provider** 2013-2016

**Veeam** Best Cloud Partner of the year (2018)

\* Netcraft 2017 -



# OVH: Our solutions



## Cloud

VPS

Public Cloud

Private Cloud

Serveur dédié

Cloud Desktop

Hybrid Cloud



## Mobile Hosting

Containers

Compute

Database

Object Storage

Securities

Messaging



## Web Hosting

Domain names

Email

CDN

Web hosting

MS Office

MS solutions



## Telecom

VoIP

SMS/Fax

Virtual desktop

Cloud HubiC

Over theBox

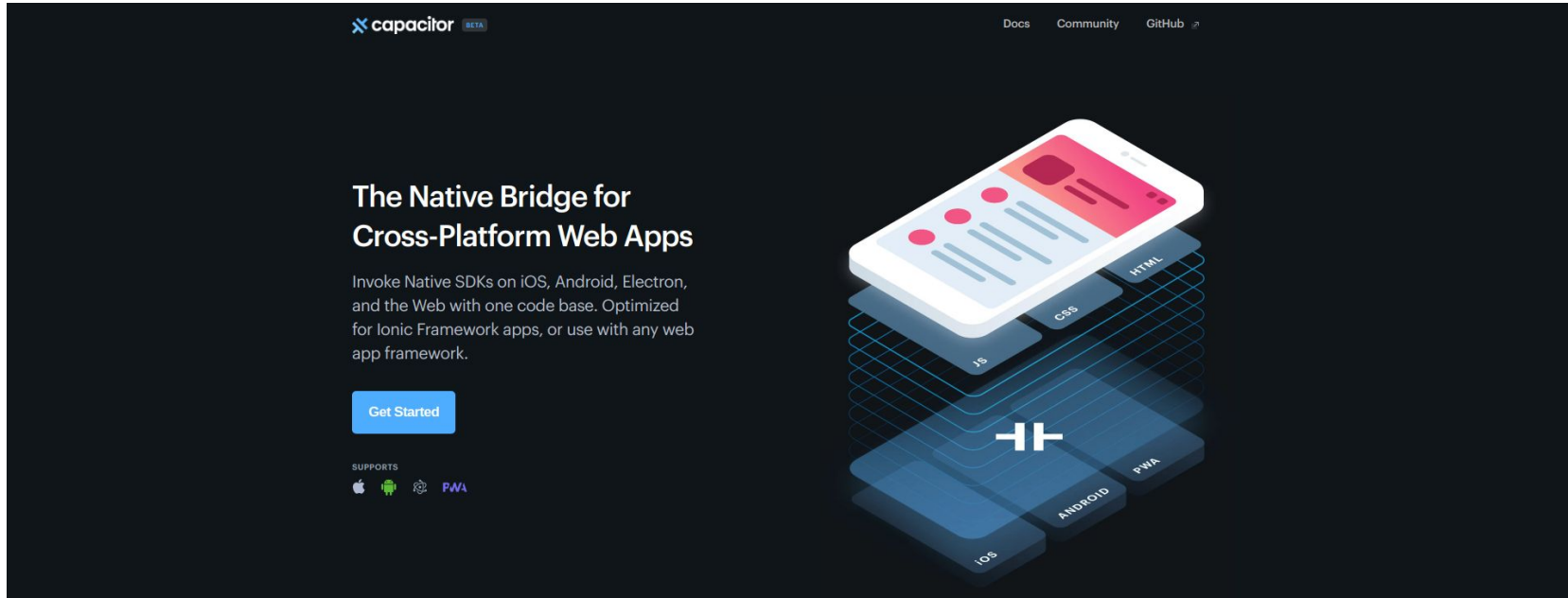


# Capacitor

The Native Bridge for  
Cross-Platform Web Apps



# What's Capacitor?



Cross-platform app runtime making it easy to build web apps that run natively on iOS, Android and web





# Spiritual heir to Apache Cordova



Evolution, not revolution

# Spiritual heir to Apache Cordova



Support for many Cordova plugins



# Extensible and evolutif

- Close to web-standards
- Plugin API
  - Swift on iOS
  - Java on Android
  - JavaScript for the web



# Developer Friendly



Easy to get started

Works on any framework



# You still need the platform tools



Android Studio and/or Xcode  
to build the native packages



# Weren't you a PWA advocate?

And you are also championing Flutter!  
Where is the coherence, guy?





# Well, I am a PWA advocate indeed



I rooted for PWA before it was fancy...

@LostInBrittany #Capacitor #DevFestLille



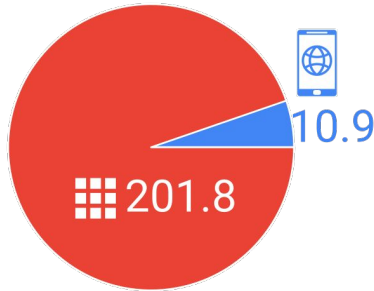
# And I know the numbers...

Top 1000 mobile apps vs top 1000 mobile web properties

■ Web mobile ■ Apps



Monthly unique visitors  
(millions)



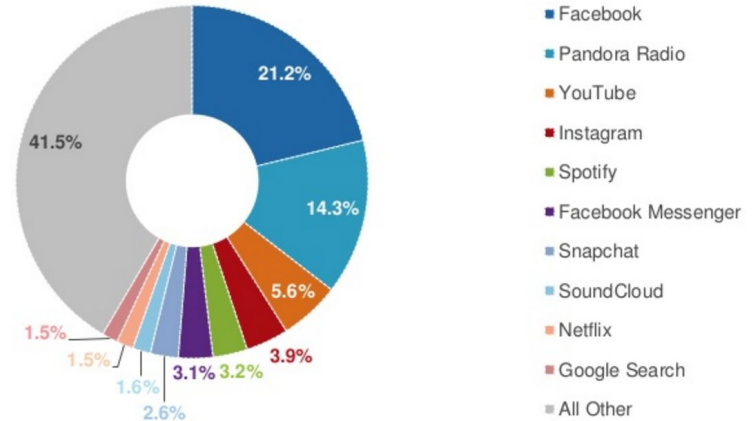
Average minutes by visitor

Source: comScore Mobile Metrix 2015, U.S., Age 18

Apps drive engagement,  
web drive visitors...

Millennials' Top Apps by Share of Total Mobile App Time Spent

Source: comScore Mobile Metrix, U.S., Age 18+, June 2015



20 biggest apps account for  
80% of user time





# An engineer role is to choose



The right tool for the right problem

# If you need super fancy UX

60 FPS, slick animations

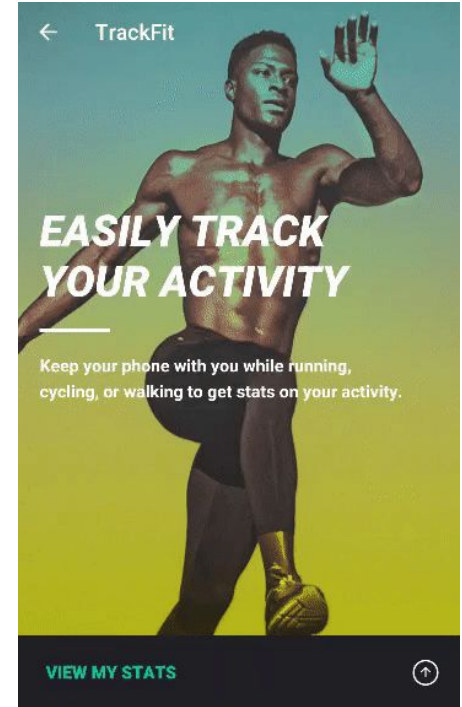
An experience more than an app



Then go **full native**

And BTW, do it in **Flutter**

believe me, you will thank me!



# For more normal needs



## California may have a huge groundwater reserve that nobody knew about

By Chris Mooney • Energy and Environment June 27

### The Washington Post

AMP helps the Washington Post increase returning users from mobile search by 23%!

"We are committed to improving speed across the board. If our site takes a long time to load, it doesn't matter how great our journalism is, some people will leave the page before they see what's there"

**David Merrell**, Senior Product Manager, The Washington Post

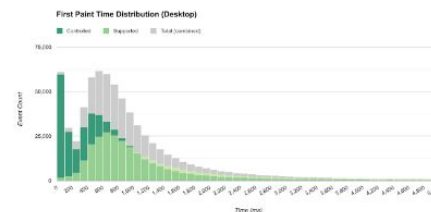
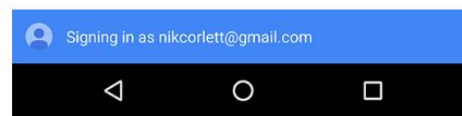
[READ MORE](#)



### AliExpress

"Smarter shopping, better living!" is the motto of AliExpress, a website where shoppers can buy everything from baby clothes to refrigerators directly from China. Part of the Alibaba Group, the global online retail marketplace is now a popular e-commerce site in America, Russia, and Brazil.

[READ MORE](#)



### Measuring the Real-world Performance Impact of Service Workers

One of the most significant benefits of service workers (from a performance perspective, at least) is their ability to proactively control the caching of assets. A web application that can cache all of its necessary resources should load substantially faster for returning visitors. But what do these gains actually look like to real users? And how do you even measure this?

[READ MORE](#)

A well done PWA is simply enough



# But if you need to be in the store?



# PWA



For many reasons, not all objective...



# Hybrid PWA apps

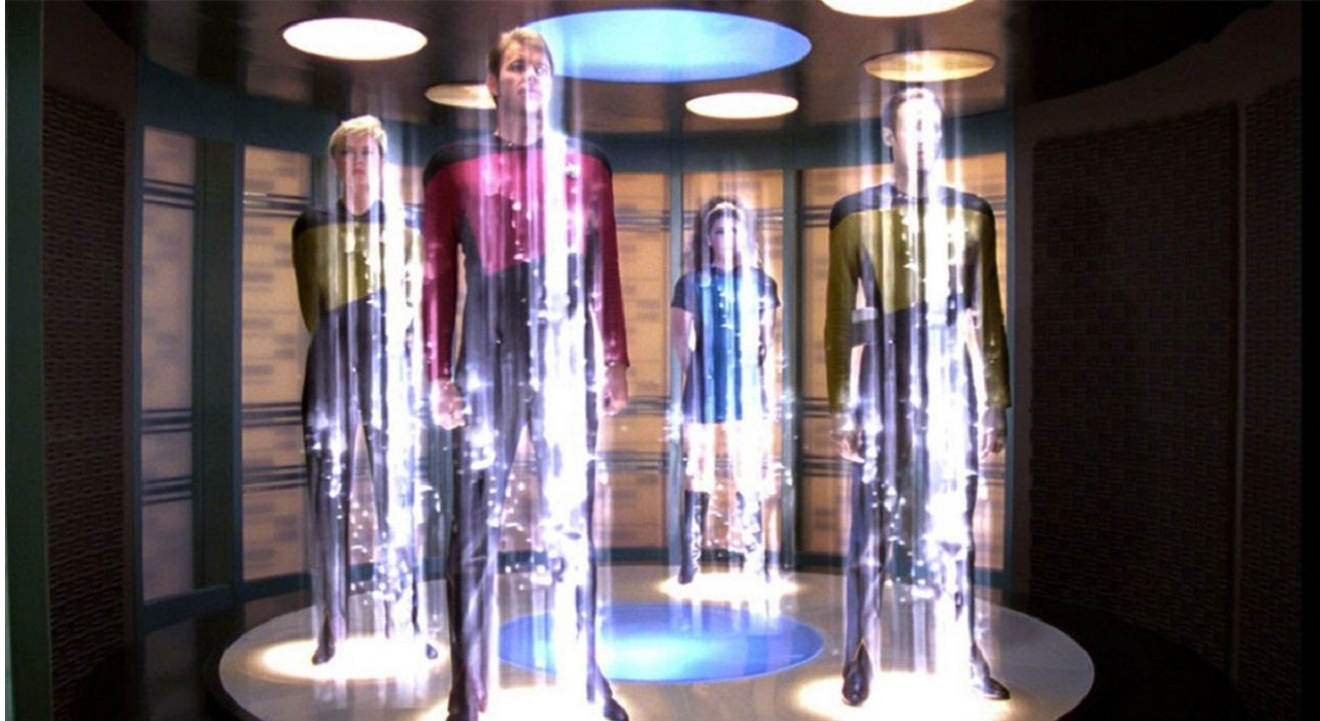


The best of two worlds





# Capacitor take your PWA to the store



In a simple, quick and painless way



# First steps testing Capacitor

Adding Capacitor to an existing app



# From web to native



Giving superpowers to you webapp



# Testing with a real webapp

Warp 10 ⚡ Photon

1.2.1

Fork me on GitHub

Backend: Insight

<https://warp10.insight.eu.metrics.ovh.net/api/v0/exec>



```
1 NEWGTS
2 256 NaN NaN NaN 9.8 ADDVALUE
3 456 NaN NaN NaN 8.7 ADDVALUE
4 656 NaN NaN NaN 7.6 ADDVALUE
5 666 NaN NaN NaN 6.5 ADDVALUE
6 686 NaN NaN NaN 9.8 ADDVALUE
7 856 NaN NaN NaN 10.9 ADDVALUE
8 1256 NaN NaN NaN 12.0 ADDVALUE
9
10 'mean' RENAME
```

▶ EXECUTE

Permalink: [#/permalink/TkVXR1RTCjI1NiBOYU4gTmFOIE5hTjA5LjggOUUREVkbEMVUUjG...](#)

Your last script execution took 427.884 µs serverside, fetched 0 datapoints and performed 46 WarpScript operations.

```
0: ▾ mean() ⓘ
  a:
  c: "mean"
  i:
  v: Array[7]
```

## Warp 10 Photon - IDE for Warp 10

@LostInBrittany #Capacitor #DevFestLille



# Step 1 - Add Capacitor to the app

- Install Capacitor

```
cd my-app
```

```
npm install --save @capacitor/core @capacitor/cli
```

- Init Capacitor

```
npx cap init
```

- Add Android and/or iOS and/or Electron support

```
npx cap add android
```



## Step 2 - Copy to Android

- Edit `capacitor.config.json` to choose the built folder
- Copy the built resources to Android

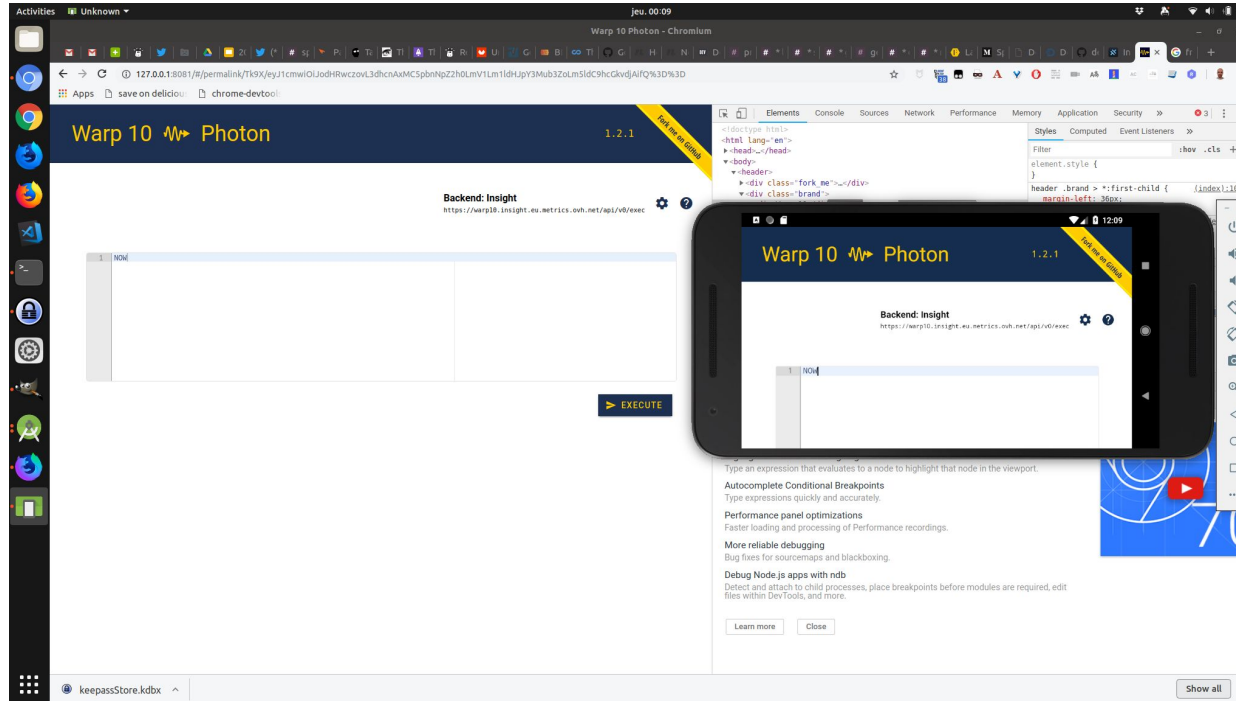
```
npx cap copy
```

- Launch Android Studio

```
npx cap open android
```



# Step 3 - Test



And our webapp is now a native app

@LostInBrittany #Capacitor #DevFestLille



# First test: successful!



Capacitor 1 - Scepticism 0



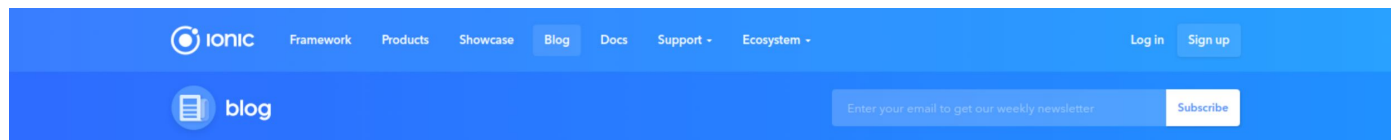
# The Capacitor example app

Nice to explore Capacitor



# Made by the Ionic team

To replace Cordova in Ionic 4



## Announcing Capacitor 1.0.0 Alpha

By [Max](#) on February 27, 2018

INTRODUCING

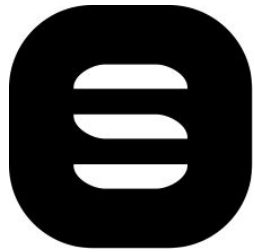


Today we are incredibly excited to announce the alpha release of a major new open source project: [Capacitor](#).



@LostInBrittany #Capacitor #DevFestLille

# Official example built on Ionic

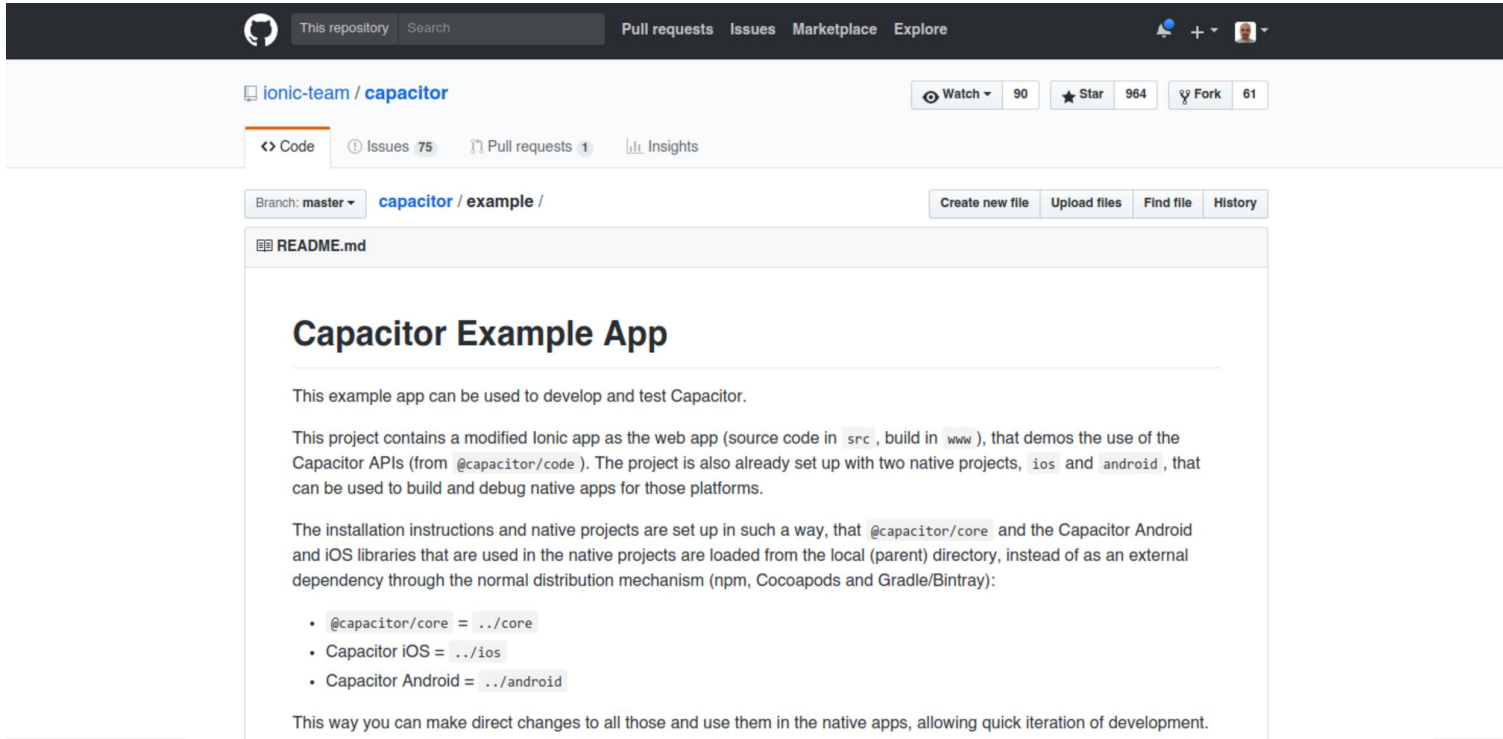


Showcasing Ionic 4





# Let's try the official example



The screenshot shows the GitHub interface for the repository 'ionic-team / capacitor'. The repository has 90 watches, 964 stars, and 61 forks. The 'example' directory is selected, showing the 'README.md' file. The README content is as follows:

## Capacitor Example App

This example app can be used to develop and test Capacitor.

This project contains a modified Ionic app as the web app (source code in `src`, build in `www`), that demos the use of the Capacitor APIs (from `@capacitor/code`). The project is also already set up with two native projects, `ios` and `android`, that can be used to build and debug native apps for those platforms.

The installation instructions and native projects are set up in such a way, that `@capacitor/core` and the Capacitor Android and iOS libraries that are used in the native projects are loaded from the local (parent) directory, instead of as an external dependency through the normal distribution mechanism (npm, CocoaPods and Gradle/Bintray):

- `@capacitor/core = ../core`
- Capacitor iOS = `../ios`
- Capacitor Android = `../android`

This way you can make direct changes to all those and use them in the native apps, allowing quick iteration of development.



# Using directly the Capacitor repository



Custom built project, not a production app

@capacitor/core and native libs loaded from local directory



# Let's build the Android version

## 1. Build Capacitor Core Module

Start by building the Capacitor Core Module in `/core`:

```
cd ../core  
  
npm install  
npm run build  
npm link
```

## 2. Build Example App

Switch back over to this example project in `/example` where you first install dependencies and link in the `@capacitor/core` you just built in the step before, then build the app and copy the build files to the correct `public` directories for both the iOS and Android example apps:

```
cd ../example  
  
npm install  
npm link @capacitor/core  
  
npm run build  
npm run copy
```

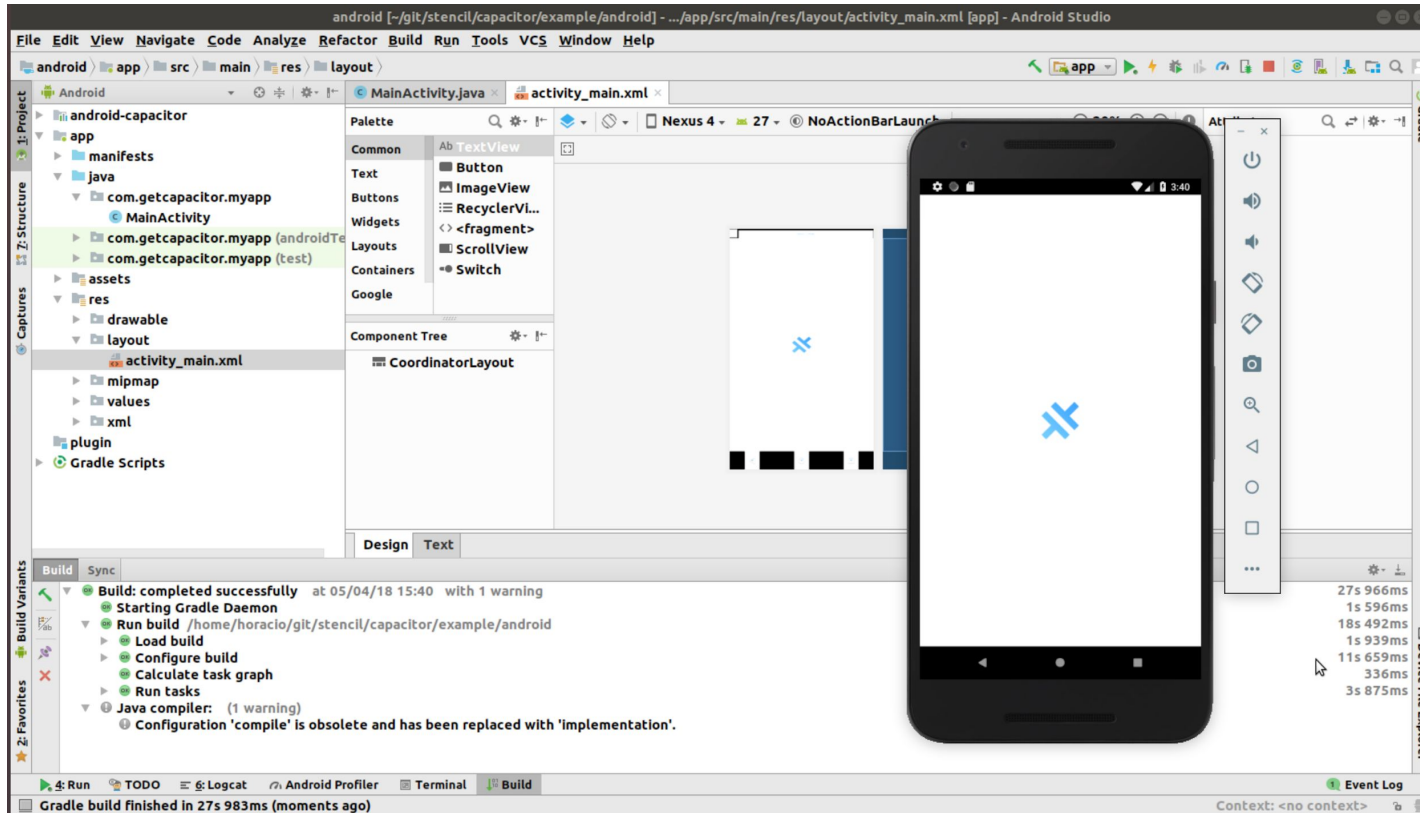
## 3. Build and run the native Capacitor Apps

Now that everything is in place you can build the native Capacitor Apps:

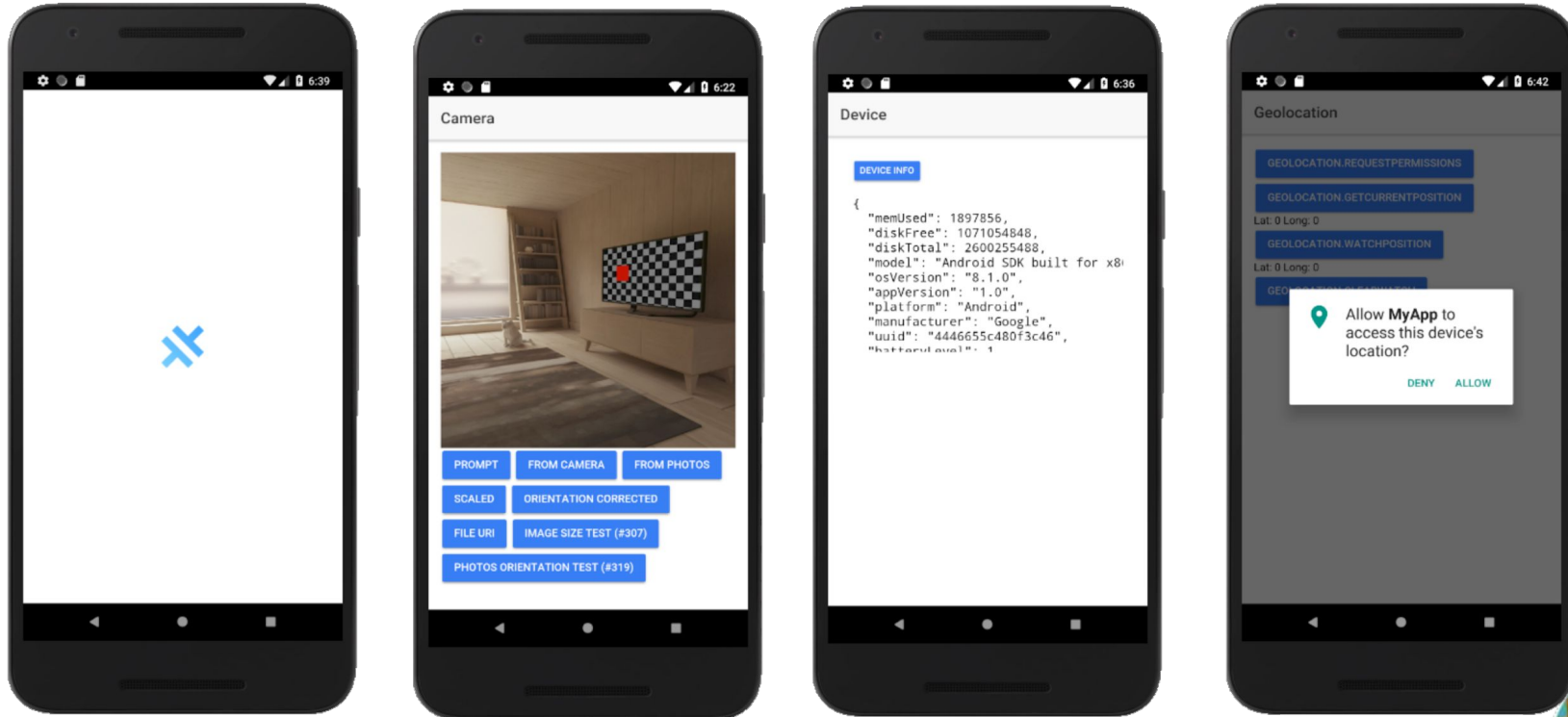
Some pre-building and building needed...



# We now have an Android project



# Easy and painless Android app



# And in PWA mode?

Toast

SHOW

Angular is running in the development mode. Call enableProdMode() to enable the production mode. [core.js:3565](#)

✖

▶Error: Uncaught (in promise): SplashScreen does not have web implementation. [web-runtime.js:40](#)

at c (polyfills.js:3)  
at Function.t.reject (polyfills.js:3)  
at CapacitorWeb.webpackJsonp.248.CapacitorWeb.pluginMethodNoop (web-runtime.js:31)  
at new MyApp (app.component.ts:40)  
at createClass (core.js:12164)  
at createDirectiveInstance (core.js:12011)  
at createViewNodes (core.js:13449)  
at createRootView (core.js:13339)  
at callWithDebugContext (core.js:14740)  
at Object.debugCreateRootView [as createRootView] (core.js:14041)

✖

▶Error: Uncaught (in promise): App does not have web implementation. [web-runtime.js:40](#)

at c (polyfills.js:3)  
at Function.t.reject (polyfills.js:3)  
at CapacitorWeb.webpackJsonp.248.CapacitorWeb.pluginMethodNoop (web-runtime.js:31)  
at new MyApp (app.component.ts:47)  
at createClass (core.js:12164)  
at createDirectiveInstance (core.js:12011)  
at createViewNodes (core.js:13449)  
at createRootView (core.js:13339)  
at callWithDebugContext (core.js:14740)  
at Object.debugCreateRootView [as createRootView] (core.js:14041)

✖

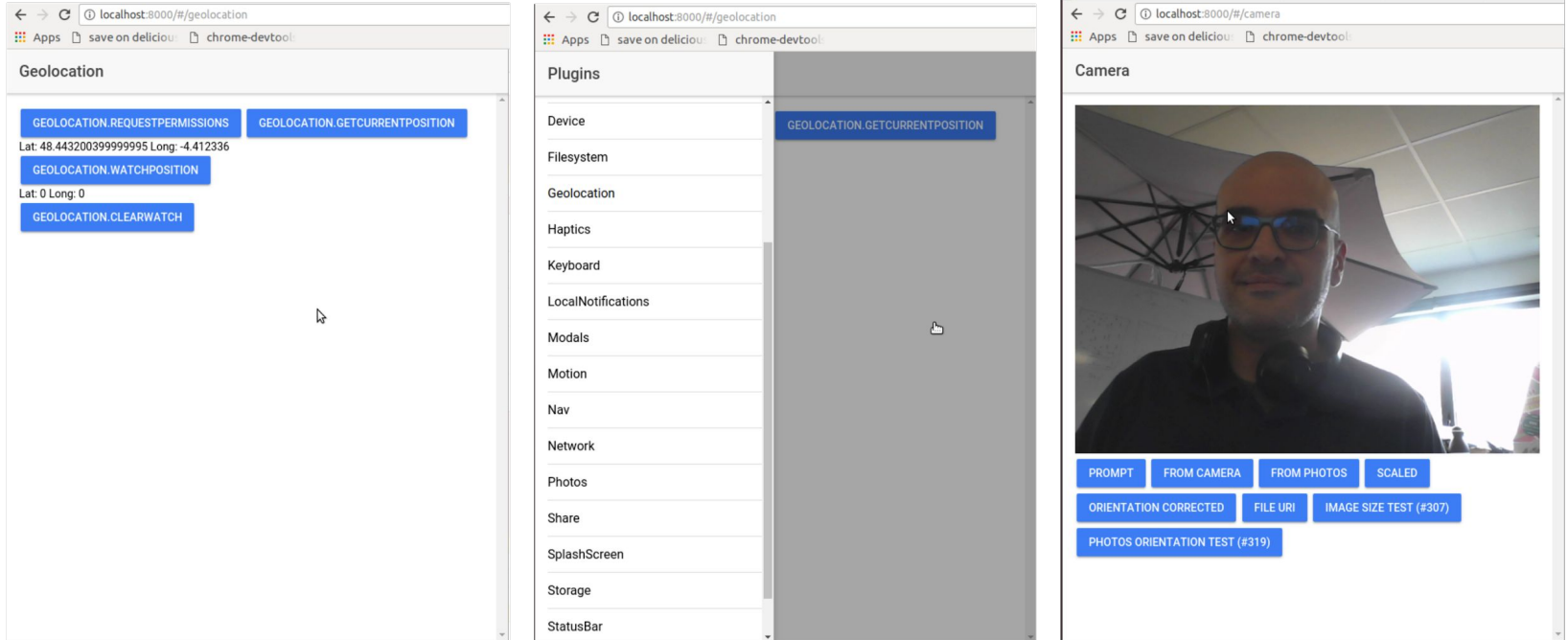
▶Error: Uncaught (in promise): App does not have web implementation. [web-runtime.js:40](#)

at c (polyfills.js:3)  
at Function.t.reject (polyfills.js:3)  
at CapacitorWeb.webpackJsonp.248.CapacitorWeb.pluginMethodNoop (web-runtime.js:31)  
at new MyApp (app.component.ts:51)  
at createClass (core.js:12164)  
at createDirectiveInstance (core.js:12011)  
at createViewNodes (core.js:13449)  
at createRootView (core.js:13339)  
at callWithDebugContext (core.js:14740)  
at Object.debugCreateRootView [as createRootView] (core.js:14041)

Some elements haven't web implementation (yet?)



# But it still works!



It fails gracefully for unsupported plugins



# Second test: successful!



Capacitor 2 - Scepticism 0





# Let's try something harder

## Stencil & capacitor



# Capacitor without Ionic



I want to use it Capacitor  
with my own toolset



# I'm a Web Components guy



And I speak a lot about Polymer

@LostInBrittany #Capacitor #DevFestLille



# But Polymer is in a transition phase



Polymer 2.x : bower based

Polymer 3: npm based but recommended only for legacy

LitElement: the new, lightweight, blazing fast library  
by Polymer team



So what to use?



Stencil, of course!

The magical, reusable web component compiler



# Let's begin with a simple example

≡ ✕ ≡ stencil & capacitor

Take a pic



A Camera app, working well on web mode

Using standard Media Capture and Streams API





# Let's charge it with Capacitor



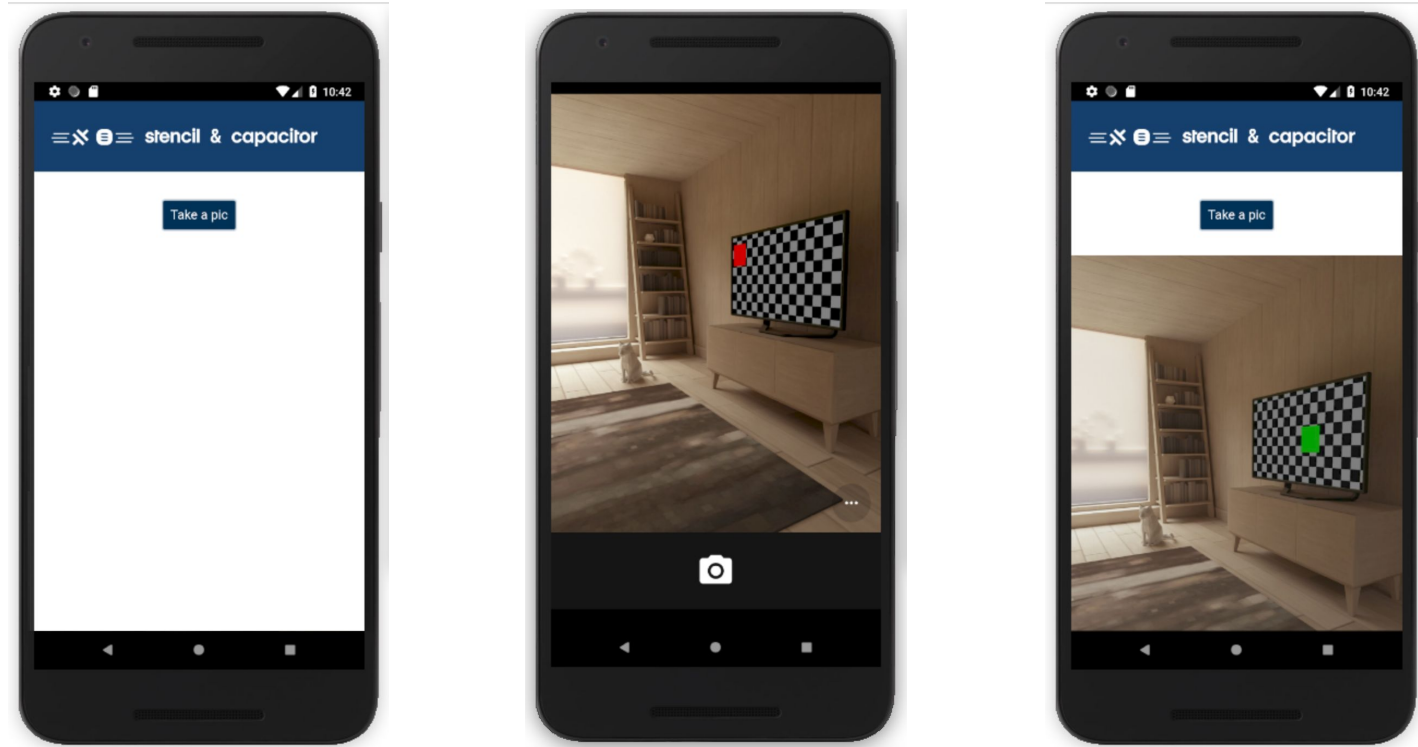
We want the same behavior

But now in Android, iOS, Electron AND web



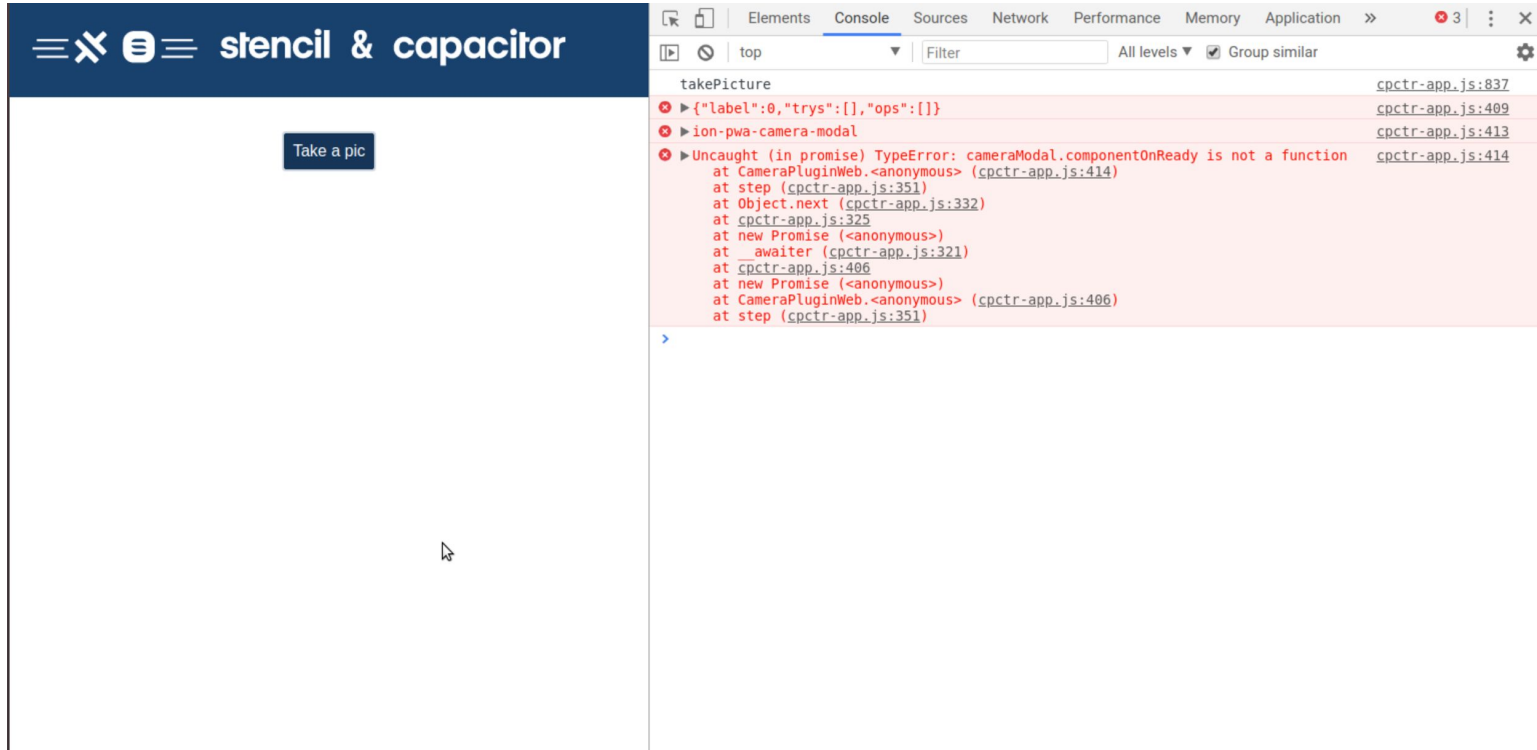


# And does it work?



Yes it does... in native mode only!

# But not in PWA mode :(



And a big question: why?



# Well, let's spot the differences...

Searching on the example code

hmmmm, [@ionic/pwa-elements](#), what's that?



# @ionic/pwa-elements

[Docs](#)[Community](#)[GitHub](#)

## Getting Started

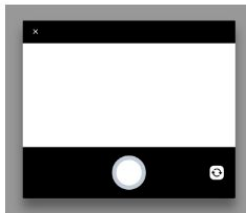
[Introduction](#)[Required Dependencies](#)[Installation](#)[PWA Elements](#)[Using with Ionic](#)

## Basics

[Development Workflow](#)[Opening Native IDE](#)[Building your App](#)[Running your App](#)[Using Cordova Plugins](#)[Native Project Configuration](#)[Progressive Web Apps](#)[iOS](#)

## PWA Elements

Some Capacitor plugins, such as `Camera`, have web-based UI available when not running natively. For example, calling `Camera.getPhoto()` will load a responsive photo-taking experience when running on the web or electron:



This UI is implemented using a subset of the [Ionic Framework](#) web components. Due to the magic of Shadow DOM, these components should not conflict with your own UI whether you choose to use Ionic or not.

## Web-based alternatives for some Capacitor plugins



# Adding @ionic/pwa-elements

Simply install them:

```
npm install @ionic/pwa-elements
```

And then import them:

```
import '@ionic/pwa-elements';
```



And then...

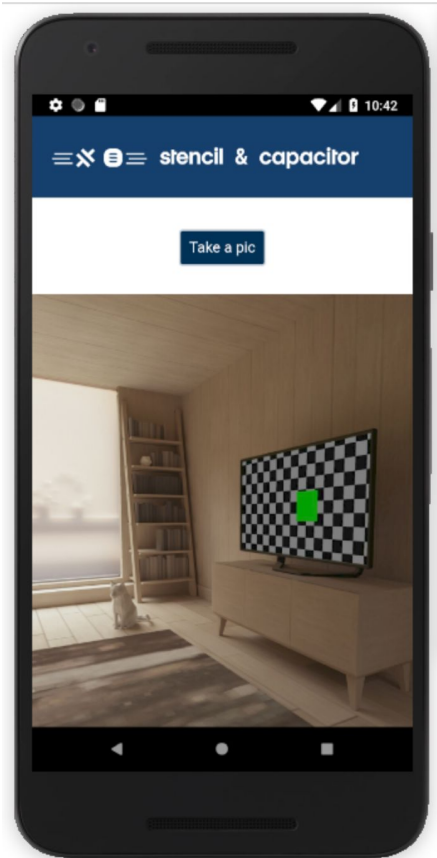


It's a kind of magic!





# On Android and Web



≡ ✕ ⓘ ≡ stencil & capacitor

Take a pic





# Second test: successful!



Capacitor 3 - Scepticism 0



# And in Real Life?

Because examples are examples...



# Use case 1: Putting PWA into store



It simply works, easy and painless!



# Use case 2: Progressive enhancing



Giving your PWA an extra P when going native



# And Capacitor already works



A true winner

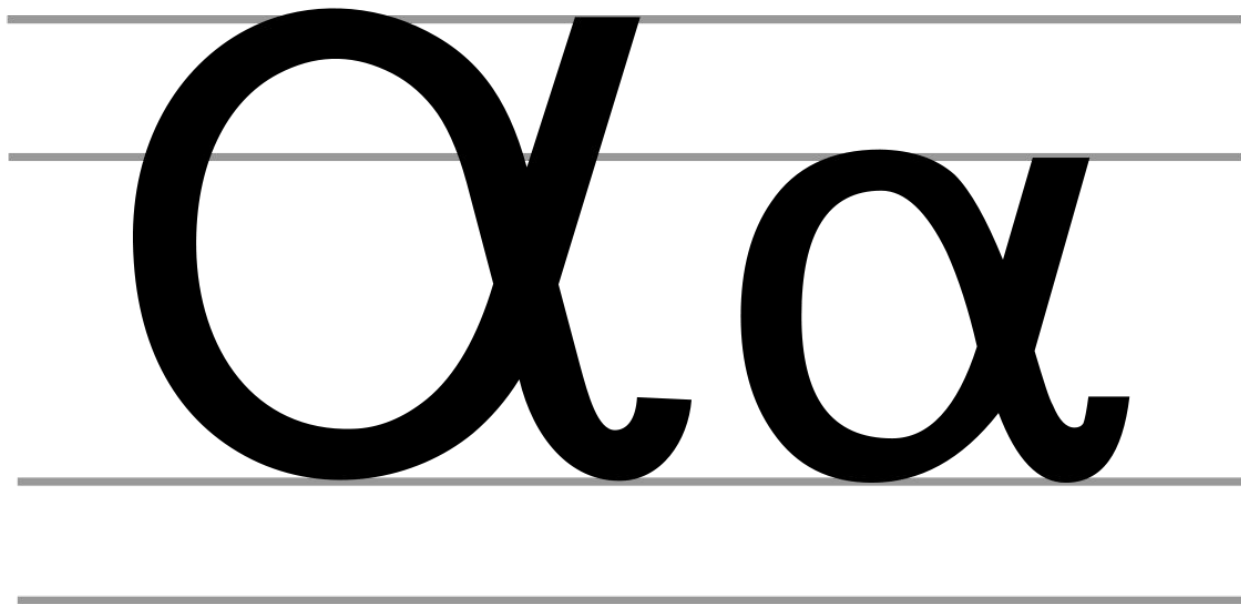


# Conclusions

## Capacitor or not?



Still fairly recent



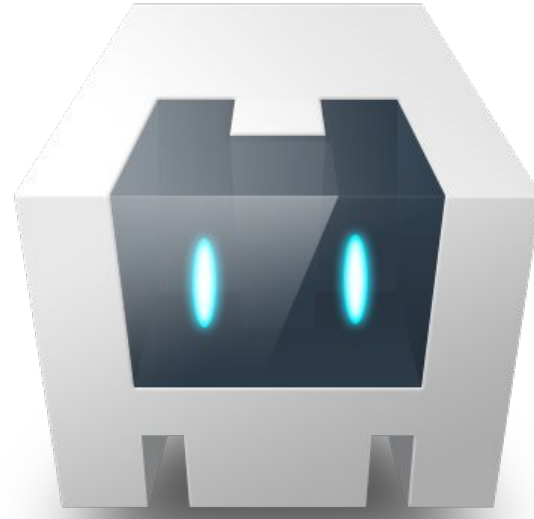
There are small glitches

Doc could be more detailed, with more examples





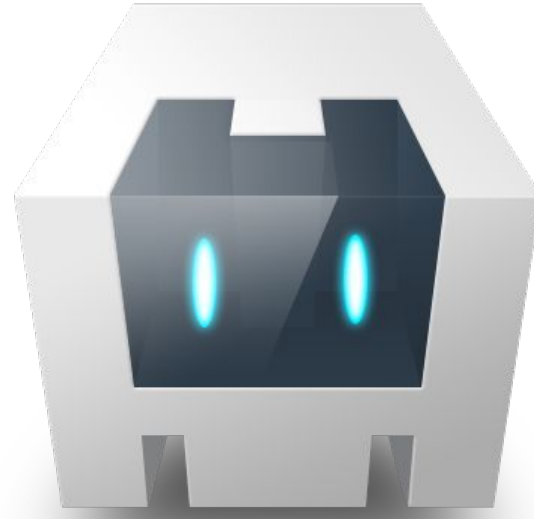
# Easy to use



Friendlier than Cordova



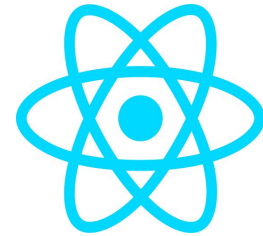
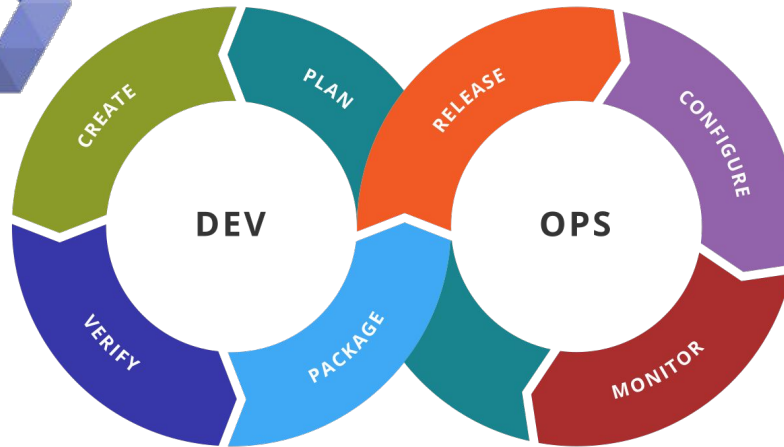
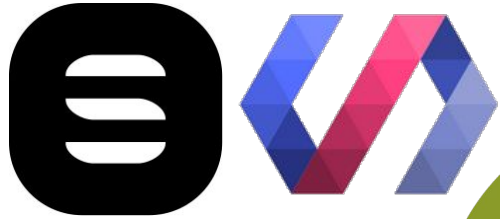
# Yet extensible



You can use existing Cordova plugins



# Not opinionated



Easy to use in any framework  
Easy to integrate in any dev toolchain



# Thank you!

